## REMARKS

Claims 1-20 remain pending in this application. No claims are amended. The title is amended herewith to correspond to the claims. The following paragraphs are numbered to correspond to the paragraph numbering in the present office action, for the Examiner's convenience.

RELATED APPLICATIONS

**1.** Paragraph [0001] is modified to provide the serial number and corresponding status of the patent application referenced at page 1 of the application. References to docket numbers are deleted.

DRAWINGS

**2.** At the initial filing date (July 28, 2001), the drawings in this application were formally filed with full compliance with 37 C.F.R. §1.84. We therefore do not understand the Examiner's comment in paragraph 2 of the present Office Action. A copy of these figures – as filed – are enclosed herewith. We kindly request reconsideration.

CLAIM REJECTIONS

**3-4.** Claims 1, 2 and 4-20 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,467,052 ("Kaler"). Applicants respectfully disagree. To anticipate a claim, Kaler must teach every element of the claim and "the identical invention must be shown in as complete detail as contained in the ... claim." *MPEP 2131* citing *Verdegaal Bros. V. Union Oil Co. of California,* 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1987) and *Richardson v. Suzuki Motor Co.,* 868 F.2d 1226, 9 USPQ2d 1913 (Fed. Cir. 1989). Kaler does not teach every element of claims 1, 2 and 4-20.

Claim 1 recites a method for processing events from electronic architecture, the architecture of the type having one or more entities generating the events, and requires the following steps:

(A) extracting the events from the architecture;

(B) separating the events according to the entities; and

(C) transforming the events to one or more text strings.

Kaler does not teach or disclose these steps. The Examiner cites Kaler, col. 3, line 66 – col. 4, line 12 with the proposition that Kaler discloses step (A). However, a close review of this section reveals that Kaler requires an in-process event creator (IEC) that *creates* an event desired by a developer. See also Kaler col. 4, lines 17-24. This is not the same as "extracting" an event previously generated by electronic architecture, as in claim 1 (and as clearly supported in Applicants' specification by, for example, the getcc processing section 102, FIG. 2).

Once Kaler generates the events, filter reduction techniques are employed to extract only information of interest from the events, to reduce the performance impact of monitoring. See Kaler, col. 4, lines 51-56. By way of example, col. 16, line 1-col. 17, line 25 of Kaler discloses how events may be predefined or custom-defined to assist in analysis (e.g., to assist in analyzing relationships between events). In particular, Kaler teaches extracting events from the software architecture to separate events according to fields, types and categories. Kaler, col. 16, line1 - col. 17, line 25.

This is not, however, the same as "extracting" events, as in step A of claim 1; nor is it the same as separating events according to entities (step B). Applicants' events are internally generated by entities of electronic architecture. In paragraph [0008] of Applicants' specification, for example, such extraction is described in the context of an extraction tool that "connects with electronic architecture to *extract* and analyze *internally generated events*" (emphasis added). There are no "event creators" as in Kaler. The events of claim 1 are preexisting (internally generated by electronic architecture, for example) and therefore have to be extracted (step A).

Consider, for example, claim 3, which depends from claim 1 and which specifically claims that the extraction of events (step A) means the extraction of "chassis logs." Chassis logs are events internally generated by connected electronic architecture and contain encoded data of electronic architecture information. See

Applicants' specification, paragraph [0003]. As such, chassis logs are for example produced by intelligent hardware modules, I/O drivers, processors (local and additional), the operating system, and system firmware; they contain information of events pertaining to operation and status of electronic architecture.

Kaler does not, at all, disclose utilization or processing of chassis logs, as in claim 3. Rather, Kaler teaches event creation and then monitoring "...the executing process for particular situations that occur which the developer wants to be monitored," for creating "an "event" that can be captured and later analyzed." Kaler, col. 4, lines 5-8. Therefore, Applicants' claim 1 (and similarly claim 3) patentably differ from Kaler in that, for example, there are no event creators; events are instead extracted from entities of electronic architecture.

Claim 1 (step B) also requires separating the events according to the entities of the electronic architecture. Kaler does not teach or suggest the separation of events according to entities of electronic architecture; rather, Kaler teaches a Local Event Concentrator (LEC) that collects and stores events of the executing process for particular situations. This LEC sends events to a developer who specifies "by means of a 'filter' what to look for in the system under examination." See Kaler, col. 4, lines 13-14. Kaler's system thus filters events according to event type, but does not separate events according to entity, as in claim 1.

For at least the reasons discussed above, Applicants contend that Kaler does not anticipate claim 1; therefore, claim 1 is patentable over Kaler. Applicants therefore respectfully request reconsideration and allowance of claim 1.

Claims 2 and 4-17 depend from claim 1 and benefit from like arguments. In addition, however, these claims provide other features that patentably distinguish over Kaler, including, for example, the following steps:

- the step of coupling a getcc extraction tool to the architecture (claim 4). Kaler is silent as to such an extraction tool. In fact, Kaler *teaches*

*away* from extraction as in step A of claim 1 by generating (not "extracting") the events with event creators. See Kaler col. 4, lines 17-24. Respectfully, we do not believe that the Examiner's statement of "telnet" in the context of Kaler's figure 1 is accurate. In particular, irrespective of modem 54 shown in figure 1 of Kaler, Kaler does not disclose, teach or suggest extracting events, as argued above in connection with claim 1.

- converting a binary representation of the events to the text strings (claim 7). Kaler does not even use the word "binary" within its text. Applicants' claim 7 thus further distinguishes over Kaler in that events internally generated may be extracted in binary form. Kaler instead creates events; Kaler's generated events appear entirely non-binary. See, e.g., Kaler Table 3, col. 17. Respectfully, the Examiner's reference to Kaler, col. 19, lines 13-22, also does not disclose binary conversion.
- analyzing the text strings and producing a human interpretable statement summarizing at least one of the events associated with the text strings (claim 8). Interpretive event summaries are not disclosed by Kaler, which instead discloses creation of custom fields. See Kaler, col. 15 and 16. Respectfully, the Examiner's reference to Kaler, col. 19, lines 50-56, does not teach producing human interpretable statements. This section in fact teaches away from claim 8 since, for example, Kaler discloses that users do not need to understand the very events generated by Kaler.
- controlling one or more steps of extracting, separating and transforming via one or more command line options (claim 10). Kaler is silent as to the use of command line options. The Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we cannot disagree more. Claim 10 patentably

distinguishes over Kaler since Kaler fails to teach command line options as in claim 10.

- controlling one or more steps of extracting, separating and transforming according to one or more configuration files (claim 11). The Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we again disagree. Claim 11 patentably distinguishes over Kaler since Kaler fails to teach configuration files as in claim 11 (the word "configuration" is not even mentioned in Kaler).
- inputting the command line options via a graphical user interface (claim 12). The Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we again disagree. Claim 12 patentably distinguishes over Kaler since Kaler fails to teach command line options and graphical user interface as in claim 12.
- updating the command line options automatically from the architecture (claim 13). The Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we again disagree. Claim 13 patentably distinguishes over Kaler since Kaler fails to teach command line options and graphical user interface as in claim 12.
- specifying one or more cells of the architecture, and extracting the events only from the one or more cells (claim 14). The Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we again disagree. Claim 14 patentably distinguishes over Kaler since Kaler fails to teach extracting events from cells of electronic architecture, as in claim 14.
- specifying one or more processors of the architecture, and extracting the events only from the one or more processors (claim 15). The

> Examiner makes a broad assertion that claims 10-20 "have been discussed in the rejection of claims 2, 4-9"; but we again disagree. Claim 15 patentably distinguishes over Kaler since Kaler fails to teach extracting events only from processors as in claim 15.

Reconsideration and allowance of claims 2 and 4-20 are thus requested.

Claim 18 is a system for processing events from electronic architecture of the type having one or more entities generating the events. Claim 18 requires the following elements:

(A) an extraction tool for extracting the events from the architecture, separating the events according to the entities, and transforming the events to one or more text strings; and

(B) an interface for coupling the extraction tool to one or more of the architecture and a log file storing the events from the architecture.

**5.** The Examiner broadly rejects claims 18-20 in paragraph 5 of the pending action without additional argument and based on previous rejections of claims 1, 2, 4-9. Respectfully we disagree since, among other reasons, Kaler does not teach the elements of independent claim 18. We have already shown and argued in connection with claim 4 that Kaler does not disclose an extraction tool, though this element is required in claim 18. Therefore, Kaler cannot anticipate claim 18. Once again, Kaler discloses generating events, not "extracting" events as required by claim 18. Reconsideration of claim 18 is requested. .

Claims 19-20 depend from claim 18 and benefit from like arguments. These claims also have other features that patentably distinguish over Kaler. For example, claim 19 discloses "chassis logs," argued above in connection with claim 3. Kaler is silent to chassis logs and cannot, therefore, anticipate claim 19. The word "chassis" is not even mentioned within Kaler. In claim 20, Kaler again does not disclose

processing events into human interpretive statements, as previously argued in connection with claim 8.

Reconsideration and allowance of claims 18-20 is thus requested and warranted.

**6-7.** Claim 3 is rejected under 35 U.S.C. §103(a) as being unpatentable over Kaler, as applied to the rejection of claims 1-2 and 6-20, in view of Applicants' admitted prior art in the background of the application (hereinafter, "background art"). Applicants respectfully disagree since, among other reasons, Kaler and the background art does not render claim 3 *prima facie* obvious.

The following is a quotation from the MPEP setting forth the three basic criteria that must be met to establish a *prima facie* case of obviousness:

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion of motivation, either in the references themselves or in the knowledge generally available to one or ordinary skill in the art, to modify the references or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP, §2142, citing *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Claim 3 describes "extracting chassis logs, wherein the step of separating the events comprises separating the chassis logs, and wherein the step of transforming events comprises transforming the chassis logs to text strings." Claim 3 depends from claim 1 and benefits from previous arguments made in connection with claim 1. Applicants have shown that Kaler does not teach each of the limitations of claim 1. We have also argued that Kaler is silent to the processing or utilization of chassis logs, as in claim 3. Therefore, Kaler alone cannot teach or suggest the limitations of claim 3, as required under 35 U.S.C. §103.

The background art also does not teach or suggest every element of Claim 3. The background art merely discloses that chassis logs are (a) generated by entities of a system during operation of the system and (b) not easily assessed, requiring an engineer intimate with the system and chassis logs to decode each chassis log to identify problems in the electronic architecture. Accordingly, Kaler and/or the background art do not each or suggest every element of Applicants' claim 3, as required under 35 U.S.C. §103. There is also no motivation to combine Kaler and the background art, since, for example, Kaler teaches generation of events and not (a) the extraction of events (claim 1) nor (b) the utilization of chassis logs from electronic architecture (claim 3). Since neither Kaler nor the background art disclose extracting events – e.g., chassis logs – from electronic architecture, as required in claim 3, claim 3 too patentably distinguishes over Kaler and the background art.

Applicants have addressed all issues raised in the Office Action dated 11 December 2003, and respectfully solicit a Notice of Allowance for claims 1-20. Applicants believe no fees are currently due, however, if any fee is deemed necessary in connection with this Amendment and Response, please charge Deposit Account No. 12–0600.

By _____

Peter C. Knops, Reg. No. 37,659
LATHROP & GAGE, L.C.
2345 Grand Boulevard, Suite 2400
Kansas City, MO 64108
Tele: (816) 460-5826
Fax: (816) 292-2001

Date: February 23, 2004